The \texttt{bropd} package*

Benjamin Tatlock\textsuperscript{1}

October 14, 2014

\textbf{Abstract}

The \texttt{bropd} package simplifies the process of writing differential operators and brackets in \LaTeX. The commands facilitate the easy manipulation of equations involving brackets and allow partial differentials to be expressed in alternate forms.

\section{Introduction}

The \texttt{bropd} package introduces a bracket command that automatically selects parentheses, square brackets or braces as appropriate based on the number of brackets. As such, brackets can be freely added and removed without alterations to the rest of an equation. Commands for typing ordinary and partial differentials are also created, with two optional concise forms for partial derivatives. Equations can therefore be written in a standard form and their style altered globally at a later date.

\section{Usage}

This section outlines the use of the three commands provided by the \texttt{bropd} package.

\subsection{Brackets}

The bracket command replaces \texttt{\left(} and \texttt{\right)} with the single command, \texttt{\br} to produce brackets in a math environment. The input \texttt{\br\{x\}} produces

\[(x).\]

When nested, the type of bracket is determined automatically, for example, \texttt{r=}\texttt{\br\{\br\{x-a\}^2+\br\{y-b\}^2\}^{-\frac{1}{2}}} results in

\[r = \left[ (x - a)^2 + (y - b)^2 \right]^{\frac{1}{2}}.\]

---

*This documentation corresponds to \texttt{bropd} v1.2, dated 2014/10/14.
\textsuperscript{1}Benjamin.Tatlock@mai1.com
2.2 Differential operators

2.2.1 Ordinary differential

The \texttt{\textbackslash od} command produces ordinary differentials, such as \texttt{\textbackslash od\{y\}\{x\}} which is displayed as \(\frac{d y}{d x}\).

The order of the differential can be specified as an optional argument, illustrated here with the command \texttt{\textbackslash od\{2\}\{u\}\{x\}=-\omega^2 u} which results in \(\frac{d^2 u}{d x^2} = -\omega^2 u\).

When using \texttt{\textbackslash od} with a sans-serif font, the package should be loaded with the optional argument \texttt{sans}.

2.2.2 Partial differential

The command \texttt{\textbackslash pd} follows the same form as \texttt{\textbackslash od}, but instead produces partial differentials. By default, \texttt{\textbackslash pd\{u\}\{t\}=6\textbackslash pd\{u\}\{x\}-\textbackslash pd\{3\}\{u\}\{x\}} produces \(\frac{\partial u}{\partial t} = 6u\frac{\partial u}{\partial x} - \frac{\partial^3 u}{\partial x^3}\).

Loading the package with the option \texttt{pdshort} results in an alternative format for partial differentials:

\[ u_t = 6uu_x - u_{xxx} \]

If the optional argument is not a positive integer, the default form is reverted to, as with \texttt{\textbackslash pd\{n\}\{x\}\{y\}}:

\[ \frac{\partial^n x}{\partial y^n} \]

A second alternative form is produced when the option \texttt{pdalt} is specified:

\[ \partial_t u = 6u\partial_x u - \partial_{xxx} u \]

The syntax for partial differentials is extended to allow higher order terms such as \(\frac{\partial^3 u}{\partial x^2 \partial t}\) to be easily typeset with \texttt{\textbackslash pd\{u\}\{x,x,t\}}. When applying a partial derivative to a longer term, the first argument can be left blank and the longer string written as an additional argument, as demonstrated by \texttt{\textbackslash pd\{\}\{z\}\{x+y\}} which results in \(\frac{\partial}{\partial z} (x + y)\).

To prevent brackets from being placed after a partial differential with an empty numerator, use \texttt{!} for the first argument, for example, \texttt{\textbackslash pd\{!\}\{x\}}:

\[ \frac{\partial}{\partial x} \]
2.3 Further examples

Combining the commands provided by this package allows more complex terms to be written concisely. The \br allows equations with multiple brackets to be constructed and manipulated quickly and easily. An example of this is \\br{x-y}\\br{Q-\\frac{1}{2}\\br{P+\\br{x-1}\\br{a+b}\\br{u+c}-\od st}}}\), which results in

\[(x - y) \left( Q - \frac{1}{2} \left\{ P + (x - 1) \left[ (A + B) (u + c) - \frac{ds}{dt} \right] \right\} \right).

The form of the partial differentials written with the \pd command can be changed at any time. \pd Et+\sum_{i=1}^3\pd{}{x_i}{\br{E+p}u_i} produces

\[\frac{\partial E}{\partial t} + \sum_{i=1}^3 \frac{\partial}{\partial x_i} [(E + p) u_i]\]

by default. With pdshort, the same input instead creates

\[E_t + \sum_{i=1}^3 [(E + p) u_i]_{x_i},\]

while with pdalt the output is

\[\partial_t E + \sum_{i=1}^3 \partial_{x_i} [(E + p) u_i].\]
3 Implementation

Announce the name, package version and \LaTeX{} requirement
\begin{verbatim}
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{bropd}
3 [2014/10/14 1.2 automated typing of brackets and differential operators]
\end{verbatim}

Initiate conditionals and counters
\begin{verbatim}
4 \newif\if@bropd@a\newif\if@bropd@b\newif\if@bropd@s\newif\if@bropd@u
5 \newcounter{@bropd@c}\newcounter{@bropd@d}
6 \newcounter{@bropd@e}\newcounter{@bropd@f}
\end{verbatim}

Provide optional arguments for alternative form partial differential and sans-serif fonts
\begin{verbatim}
7 \DeclareOption{pdshort}{\@bropd@atrue\@bropd@ufalse}
8 \DeclareOption{pdalt}{\@bropd@utrue\@bropd@utrue}
9 \DeclareOption{sans}{\@bropd@strue}\ProcessOptions
\end{verbatim}

Define character for $\pd$ with empty numerator and no brackets following
\begin{verbatim}
\def\@bropd@t{!}
\end{verbatim}

\begin{verbatim}
\@bropd@g
\end{verbatim}

Internal command for default partial differential form
\begin{verbatim}
\def\@bropd@g#1#2#3#4{
\def\@bropd@h{#1}
\ifx\@bropd@o\@bropd@t\let\@bropd@o\empty\fi
\resetcounters
\ifx\@bropd@h\empty\let\@bropd@h\empty\fi
\setcounter{@bropd@c}0\setcounter{@bropd@d}1
\let\@bropd@h\empty
\def\@bropd@o{#4}
\ifnum\value{@bropd@c}>1\^\arabic{@bropd@c}\fi
\setcounter{@bropd@d}1
\let\@bropd@h\@bropd@i
\@for\@bropd@i:=#3\do{
\ifx\@bropd@i\@bropd@h
\stepcounter{@bropd@d}\else\@bropd@h\ifnum\value{@bropd@d}>1\^\arabic{@bropd@d}\fi#4\setcounter{@bropd@d}1\fi
\let\@bropd@h\@bropd@i}
\@bropd@h
\end{verbatim}

\begin{verbatim}
\def\@bropd@t{!}
\end{verbatim}

Create numerator
\begin{verbatim}
\ifx\@bropd@o\@bropd@t\let\@bropd@o\empty\fi
\resetcounters
\ifx\@bropd@h\empty\let\@bropd@h\empty\fi
\setcounter{@bropd@c}0\setcounter{@bropd@d}1
\let\@bropd@h\empty
\def\@bropd@o{#4}\ifnum\value{@bropd@c}>1\^\arabic{@bropd@c}\fi
\setcounter{@bropd@d}1
\let\@bropd@h\@bropd@i
\@for\@bropd@i:=#3\do{
\ifx\@bropd@i\@bropd@h
\stepcounter{@bropd@d}\else\@bropd@h\ifnum\value{@bropd@d}>1\^\arabic{@bropd@d}\fi#4\setcounter{@bropd@d}1\fi
\let\@bropd@h\@bropd@i}
\@bropd@h
\end{verbatim}

Cycle through each term for the denominator. If a term is the same as the previous, add to counter, otherwise print along with counter value
\begin{verbatim}
{\@for\@bropd@i:=#3\do{
\ifx\@bropd@o\@bropd@i\let\@bropd@o\empty\fi
\setcounter{@bropd@d}1
\else
\setcounter{@bropd@d}0
\ifnum\value{@bropd@d}>1\^\arabic{@bropd@d}\fi
\fi
#4\setcounter{@bropd@d}1
\let\@bropd@h\@bropd@i}
\@bropd@h
\end{verbatim}
If optional argument provided, display partial differential to stated order
\frac{\#4^{\#1}\@bropd@o}{\noexpand\#3^{\#1}}

\@bropd@j Internal command for brackets. Evaluate terms within brackets to determine the highest bracket level
\def\@bropd@j#1{\setcounter{@bropd@f}0
\begingroup\renewcommand{\br}[1]{\stepcounter{@bropd@e}##1
\ifnum\value{@bropd@e}>\value{@bropd@f}
\setcounter{@bropd@f}{\value{@bropd@e}}
\fi
\addtocounter{@bropd@e}{-1}}\sbox0{$#1$}\endgroup
\loop
\ifnum\value{@bropd@f}>2
\addtocounter{@bropd@f}{-3}
\repeat

Determine bracket type for current level
\ifnum\value{@bropd@f}>0
\ifnum\value{@bropd@f}>1
\def\@bropd@k{\left\lbrace#1\right\rbrace}
\else
\def\@bropd@k{\left\[#1\right\]}
\fi
\else
\def\@bropd@k{\left(#1\right)}
\fi}

\@bropd@l External command for brackets
\newcommand\br[1]{\let\@bropd@m\@bropd@l
\@bropd@j{\let\@bropd@l\empty#1\let\@bropd@l\@bropd@m}
\ifx\@bropd@l\empty
\def\@bropd@n{\@bropd@k}
\else
\let\@bropd@m\@bropd@l
\def\@bropd@n{\@bropd@k_{\@bropd@m}}
\fi
\let\@bropd@l\empty
\@bropd@n}

\@bropd@q Internal command for first alternative partial differential command
\def\@bropd@q#1#2#3{\def\@bropd@h(#1)
\ifx\@bropd@h\empty

Default partial differential if character provided as optional argument

\ifnum9<1!\else_\fi
\else
\@bropd@true
\fi
\ifx\@bropd@o\@bropd@t
\@bropd@true\let\@bropd@o\empty
\fi
\if\@bropd@b
\@bropd@g{#1}\@bropd@o{\noexpand#3}\partial
\else
Construct alternate form partial differential
\partial_{\ifx\@bropd@h\empty
\@for\@bropd@i:=#3\do\@bropd@i\else\setcounter{@bropd@c}0
\loop
\ifnum\value{@bropd@c}<#1
#3\stepcounter{@bropd@c}
\repeat
\fi}
\@bropd@o
\fi
\else
Internal command for second alternative partial differential command
\def\@bropd@q#1#2#3{\def\@bropd@h{#1}
\ifx\@bropd@h\empty
\@bropd@false
\else
Default partial differential if character provided as optional argument
\ifnum9<1!\else_\fi
\else
\@bropd@false
\fi
\else
\@bropd@true
\fi
\fi
\if\@bropd@o\@bropd@t
\@bropd@true\let\@bropd@o\empty
\fi
\if\@bropd@b
\@bropd@g{#1}\@bropd@o{\noexpand#3}\partial
\else
Construct alternate form partial differential
\def\@bropd@p{\ifx\@bropd@h\empty
\@for\@bropd@i:=#3\do\@bropd@i\else\setcounter{@bropd@c}0
\loop
\ifnum\value{@bropd@c}<#1
#3\stepcounter{@bropd@c}
\repeat
\fi}
\#3\stepcounter{\bropd@o}
\repeat \fi
\ifx\bropd@o\empty
\let\bropd@l\bropd@p\else\bropd@o_{\bropd@p}\fi\fi
\fi
\else
Internal command for default partial differentials
\def\bropd@q#1#2#3{\bropd@g{#1}{#2}{\noexpand#3}\partial}
\br External command for brackets
\newcommand\br[1]{\bropd@j{#1}\bropd@k}
\pd
\newcommand\pd[3][]{\def\bropd@h{#1}\frac{\bropd@r\ifx\bropd@h\empty\else^{#1}\fi#2}{\bropd@r#3\ifx\bropd@h\empty\else^{#1}\fi}}
\od Format ordinary differential based on font type
\if\bropd@s
\def\bropd@r{\mathsf d}
\else
\def\bropd@r{\mathrm d}
\fi
\ed External command for ordinary differentials
\newcommand\od[3][]{\def\bropd@h{#1}\frac{\bropd@r\ifx\bropd@h\empty\else``#1\fi\#2}{\bropd@r\#3\ifx\bropd@h\empty\else``#1\fi}\}}
\endinput
Change History

v1.0
General: Initial version ........... 4

v1.1
\bropd@g: Fixed higher order differ- 
ential with denominators         \bropd@q: Added second alternative form for partial differ- 
containing commands ........... 5  
\bropd@q: Added second alternative form for partial differentials ........... 5  
Allowed partial differential with

empty numerator without need
for bracket command after .... 6
\pd: Brackets automatically added
after empty numerator ....... 7

v1.2
\bropd@j: Changed \bgroup to
\begingroup and \egroup to
\endgroup to remove erroneous introduction of space before and
after brackets ............. 5